基于标记变迁系统的可信计算平台信任链测试

徐明迪 张焕国 严 飞
（空天信息安全与可信计算教育部重点实验室 武汉 430072）
（武汉大学计算机学院 武汉 430079）
（武汉大学软件工程国家重点实验室 武汉 430072）

摘 要 可信计算是当今世界信息安全领域的重要潮流之一. 根据国家有关规定，信息安全产品需要经过测评认证，但目前国内外对可信计算测试的理论与技术研究还非常不完善，也无相应测试工具或系统，这必然影响可信计算的发展. 该文着眼于规范定义的信任链行为特征，以进程代数作为指称语义描述工具，以标记变迁系统作为操作语义，对规范定义的信任链行为特征进行了形式化描述，提出了一种基于标记变迁系统的信任链测试模型框架. 针对信任链规范与实现之间的问题，从易测性出发对测试集进行了有效约简；并论证了信任链的规范实现与规范说明之间的关系，为测试用例构造方法提供了理论依据，从而解决了信任链测试这一难题.

关键词 可信计算平台；信任链；进程代数；标记变迁系统；一致性测试
中图法分类号 TP309 **DOI 号**：10.3724/SP.J.1016.2009.00635

Testing on Trust Chain of Trusted Computing Platform Based on Labeled Transition System

XU Ming-Di ZHANG Huan-Guo YAN Fei
(*Key Laboratory of Aerospace Information Security and Trust Computing of Ministry of Education*, *Wuhan* 430072)
(*School of Computer*, *Wuhan University*, *Wuhan* 430079)
(*State Key Laboratory of Software Engineering*, *Wuhan University*, *Wuhan* 430072)

Abstract At present, the trusted computing has become a major development trend in the field of information security. According to "China National Information Security Certification and Evaluation Management", the information security products must be evaluated and certified. However, attribute to the situation that the specifications are established by the enterprises, the theory about the trusted computing is behind the technology, and there aren't any testing tools, which will affect the development of trusted computing. Therefore, this paper focuses on the behavior characters of specifications of trust chain, letting process algebra as denotation semantics, and taking labeled transition system (LTS) as operation semantics for formalizing behavior characters of trust chain. The authors put forward a test framework based on LTS for trust chain. Aiming at problems between specifications and implementations of trust chain, they reduce test suite effectively for testability, and reason the relationship of them, which are theory foundation of constructing test suite and solve the hard problem of trust chain test.

Keywords trusted computing platform; trust chain; process algebra; labeled transition system; conformance testing

---

收稿日期：2008-12-08；最终修改稿收到日期：2009-01-17. 本课题得到国家"八六三"高技术研究发展计划项目基金（2007AA01Z411，2008AA01Z404）、国家自然科学基金（60673071，90718005，90718006）和空天信息安全与可信计算教育部重点实验室开放基金资助.
徐明迪，男，1980 年生，博士研究生，主要研究方向为可信计算与系统安全. E-mail：mingdixu@163.com. **张焕国**（通信作者），男，1945 年生，教授，博士生导师，主要研究领域为信息安全与可信计算. E-mail：liss@whu.edu.cn. **严 飞**，男，1980 年生，讲师，主要研究方向为可信计算.

1 引 言

21 世纪是信息的时代，在信息安全的实践中，人们逐渐认识到，大多数安全隐患来自于微机终端，因此必须确保源头微机的信息安全. 而这必须从微机的芯片、硬件结构和操作系统等方面综合采取措施. 由此产生出可信计算的基本思想[1].

1999 年，IBM、HP、Intel 和微软等著名 IT 企业发起成立了可信计算平台联盟 TCPA(Trusted Computing Platform Alliance). 2003 年 TCPA 改组为可信计算组织 TCG(Trusted Computing Group). TCPA 和 TCG 已经制定了关于可信计算平台、可信存储和可信网络连接等一系列技术规范①，随之相关的可信计算技术和应用领域也逐渐步入市场.

可信计算的基本思想是[1]：首先构建一个信任根，再建立一条信任链，从信任根开始到硬件平台，到操作系统，再到应用，一级认证一级，一级信任一级，把这种信任扩展到整个计算机系统，从而确保整个计算机系统的可信. 信任根和信任链是可信计算平台的最主要的关键技术之一. 信任链把信任关系从信任根扩展到整个计算机系统. 在 TCG 的可信 PC 技术规范中，给出了可信 PC 信任链的实现方法②.

作为一种由技术需求推动的安全产品，可信计算领域存在着“技术领先，理论滞后”，“理论落后于技术”的现状[2]. 对于可信计算的理解，在学术界也尚未形成统一性的意见，可信计算平台的测试理论研究在国内外尚属空白. 同时，由于可信计算平台的特殊性，可直接用于借鉴的理论工具也并不多，这就加大了对可信计算平台测试理论的研究. 但是，可信计算产品已经是一种实际存在的安全产品，从社会需求来看，可信计算平台将会得到广泛应用. 但是根据国家有关规定，安全产品的实际应用是需要进行相应的测试评估的，因此可信计算平台的测试理论研究是一个迫在眉睫的问题.

一般来说，规范通常是采用自然语言或者非形式化语言的一种描述，规范本身的高度抽象和二义性使得实际工程通常并未完全遵照规范说明来实现. 可信计算平台信任链规范也不例外，信任链是计算系统的一个组成部分，它是伴随着系统的运行而存在的，这就导致了信任链不是以整体形式实现的软件，如文献[3]将信任链从 BIOS 延伸到操作系统内核层，由于信任链功能上分散的特性，加上不同厂商对可信计算规范理解上存在的差异以及面对市场所做出的妥协，出现了产品实现并未完全遵循规范说明的现象，规范说明与系统实现之间的鸿沟使得难以有效地对同一规范下的不同可信 PC 产品进行测试和评估，这将对可信计算产品的安全性造成潜在影响.

同时，信任链是可信计算保障系统安全可靠的主要技术手段，信任链和系统之间是一种相互依赖且并存的关系. 但是，如果信任链的实现都未遵从规范说明，那么系统的安全性也就无法得到基本保障，可以说，信任链实现程度的好坏直接影响着系统的安全性. 因此，如何针对可信 PC 的特征，提出一种信任链测试机制，实现对同一规范下的不同产品进行有效测试，并通过测试结果对被测对象进行有效的安全评估，成为当前可信 PC 信任链测试所面临的一个主要难题.

从计算机系统的角度出发，信任链和可信计算平台之间是一种交互和并行的关系，交互是指信任链需要在特定阶段获得系统的软硬件信息并生成完整性度量并以此为基础提供完整性报告，这种交互关系是通过规范说明体现的；另一方面，从宏观角度来看，信任链和系统之间保持着工作方式上的并行关系，信任链完成信任传递的任务，而系统则完成正常启动的工作.

为此，本文针对可信计算中信任链规范一致性测试缺乏理论模型的问题，使用进程代数作为形式化的工具描述信任链，对信任链的行为特征进行推导，从动态的角度建立了基于进程代数的信任链测试模型，并运用标记迁移系统对信任链规范进行形式化说明，从系统易测试的角度归纳信任链测试所包含的基本测试集，使用该测试集对可信计算平台信任链进行一致性测试.

2 相关工作

安全测试既包括直接、明确的测试方法，也包括程序繁杂且持续时间长的重复性测试. 我们把这种

① Trusted Computing Group. TCG Specification Architecture Overview [EB/OL]. 2005-03-01. https://www.trustedcomputinggroup.org

② Trusted Computing Group. PC Client Specifications, Version 1.2 Revision 1.00 [EB/OL]. 2005-07-13. https://www.trustedcomputinggroup.org/specs/PCClient

按照规定的程序，为给定的产品、设备、组件、现象、工艺或服务的一种或多种特性的技术操作定义为一致性测试[4].

目前一致性测试更多集中在对协议的测试上，协议一致性测试经过 20 来年的发展，取得了很大的进展. 国际标准化组织 ISO 制定了国际标准[5]，给出了一致性测试的概念、方法和框架等等. 协议一致性测试的研究工作主要包括两方面：测试生成和测试执行. 测试生成通过分析协议说明来确定要测试的各个方面，从而产生测试套（测试例的集合）. 测试执行部分为测试例的运行提供环境，并分析测试结果，给出测试判决[6].

我们在前面提到，信任链不是以整体形式实现的软件. 信任链规范不单纯是一个软件规范，也不仅仅是一个协议规范，它包含了（1）与系统进行交互的流程；（2）逻辑上并行的信任链建立规则；（3）软件功能接口. 对于信任链的软件功能接口（如可信平台模块 TPM 驱动程序以及对外服务例程）的测试可以采用软件测试的方法，如基于规范的功能测试方法和基于程序的结构测试方法[7]，但对于（1）和（2）的测试，就需要一种形式化的方法去描述信任链和系统之间的交互过程，并能够刻画出信任链和系统之间并行工作的特性，也就是观察在外部（系统）或者内部动作的激励下，信任链所响应的输出.

对于并发系统的行为刻画，标记变迁系统 LTS（Labeled Transition System）是一种传统的被广为接受的基于抽象状态机的状态变迁模型. 标记变迁系统最先分别由 Keller[8] 和 Lien[9] 两人提出来，而后 Plotkin 在此基础上将变迁系统中增加了一个元：标记（动作）[10]. LTS 是形式化一致性测试的一个重要研究方向[11]，它常常用于形式化地描述分布式系统和协议的行为，成为这些系统说明、实现和测试研究的有力工具，同时也可以对安全协议进行验证[12]. 另外，LTS 还能作为许多形式描述语言的语义模型，如 CCS、CSP、LOTOS、ESTELLE、SDL 等. 文献[6]使用 LTS 进行测试集自动生成的研究，文献[13]采用扩展 LTS 进行了分布式系统一致性测试的研究.

3　信任链测试理论

3.1　信任链规范说明模型

从功能性来说，信任链规范着重定义了信任链的建立流程以及为了建立信任链，外部实体需要访问 TPM API 的接口；从安全性来说，信任链规范规定了客体对象访问 TPM 的规则，如 Locality（访问通道）使用原则，访问 TPM 内部 NV（非易失性）存储区域的方法. 图 1 描述了信任链规范说明中的功能说明和安全说明. 可以看出，该规范所包含的内容非常丰富，根据我们掌握的可信计算平台测试经验，不同可信 PC 产品在规范实现上的差异导致了单纯地从规范说明的角度对其进行逐一测试的局限性.

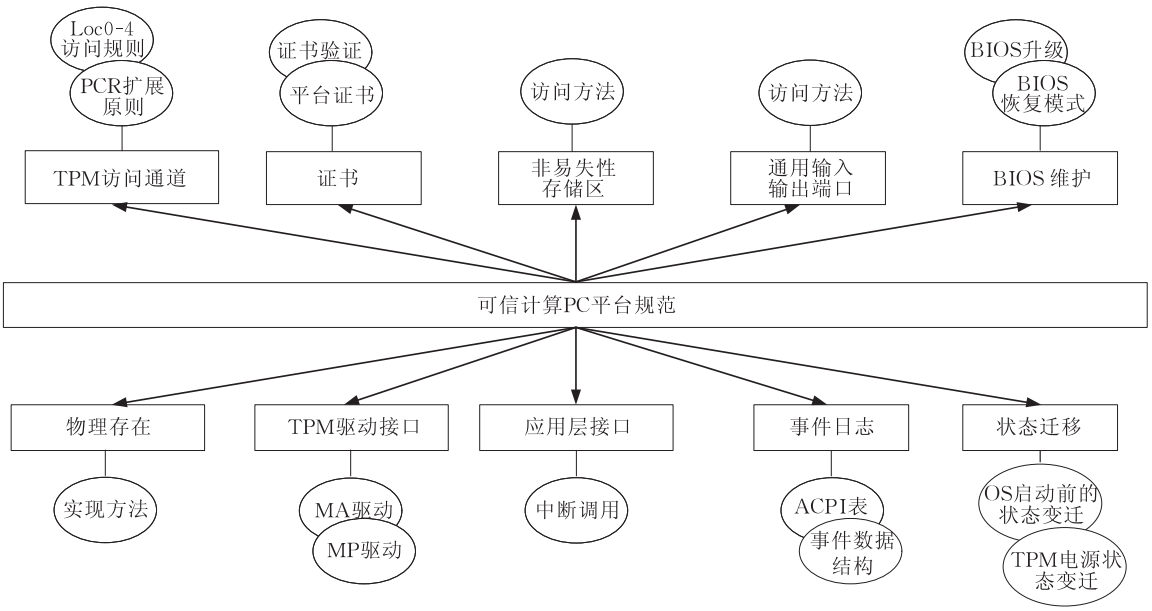

图 1　可信计算 PC 规范说明

从图 1 中可以看出，信任链规范主要定义了两种对象：（1）TPM 访问规则对象；（2）信任链建立对象. 为了能够抽象出信任链规范中的角色进程，建立不同角色之间的交互关系，我们将可信计算平台抽

象为 3 个实体：可信平台模块(TPM)、可信度量根(RTM)、由硬件和软件所组成的系统(System)，如图 2 所示.

我们对图 2 进行如下说明：

(1) RTM 通过通道度量 System 的完整性，并记录 System 的行为特征；

(2) RTM 通过 Locality 访问 TPM，并将(1)得到的完整性信息扩展至平台配置寄存器(PCR)；

(3) RTM 将控制权交给 System，System 开始运行.

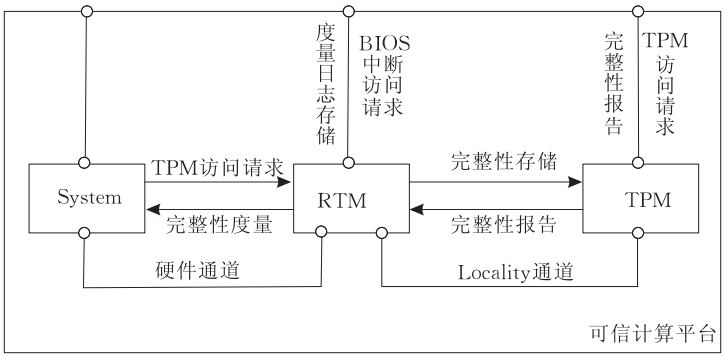

图 2 可信计算 PC 规范说明模型

以上 3 步是信任链建立的主要思想，同时也是可重复的，在信任链运行期间，TPM、RTM 和 System 之间都是存在交互的，这种交互影响可信计算平台的运行状态在不断地变迁着，下面我们给出信任链建立的形式化说明.

Booting$(m)\equiv$log $m.pb_loc$;

extend $m.PCR(s)$, $m.pb_loc$, $m.locality$;

$p=$read $m.pb_loc$;

call p;

log $m.bl_loc$;

extend $m.PCR(s)$, $m.bl_loc$, $m.locality$;

$b=$read $m.bl_loc$;

call b;

log $m.osk_loc$;

extend $m.PCR(s)$, $m.osk_loc$, $m.locality$;

$o=$read $m.osk_loc$;

call o;

其中，m 表示可信计算平台，它包含了前面提到的 System、RTM 和 TPM，log 表示 RTM 建立日志记录，extend 表示将内存数据 $m.*_loc$ 的完整性度量值通过 $m.locality$ 通道扩展到 PCR 寄存器中，read 表示读取代码，call 表示执行代码[14]. 这里，我们就可以将图 1 所描述的信任链规范建模为图 2 中 System、RTM 和 TPM 3 个实体之间的协议系统，一个可信计算平台的信任链规范就能够通过交互协议(或称为规则)来进行描述.

3.2 LTS 的相关理论

一个 LTS 由状态以及带标记的状态间的变迁组成. LTS 是基于状态和变迁的抽象关系模型. 任何一个给定的模型，给出全局状态和原子动作及对应的状态变迁规则，就可以定义出相应的变迁系统. 许多并发系统的属性都可以用变迁系统进行描述.

定义 1. 标记变迁系统(Labeled Transition System)是一个四元组$(Q,A,\xrightarrow{\mu},q_0)$：$Q$ 是非空有限状态集合；A 是有限的基本动作集合；$\mu\in A\cup\{\tau\}$，$\xrightarrow{\mu}$是集合 Q 上的二元关系；$q_0\in Q$ 是初始状态.

基本动作(也称为标记)集合 A 表示系统所有的可观察动作，特殊的标记 τ 表示外部不可观察的内部动作. $\xrightarrow{\mu}$中的元素是一个三元式$(q,\mu,q')\in\xrightarrow{\mu}$，其中，$q,q'\in Q$ 且 $\mu\in A\cup\{\tau\}$也可以写成 $q\xrightarrow{\mu}q'$，表示系统在动作 μ 下从状态 q 到达状态 q'，而 $q\xrightarrow{\tau}q'$表示系统从状态 q 到达状态 q'，执行动作是外部不可见的. 一个标记变迁系统也可以用树来描述，树的根节点是初态 q_0，变迁关系用边和两端的节点 $A\cup\{\tau\}$表示，不同的节点表示不同的状态[15]. 将 A 上的所有标记变迁系统记为 $LTS(A)$.

定义 2. $T=(Q,A,\xrightarrow{\mu},q_0)$，$q\in Q$，那么

(1) $Init(q)=\{a\in A\mid q=a\Rightarrow\}$；

(2) $Trace(q)=\{\sigma\in A^*\mid q=\sigma\Rightarrow\}$；

(3) $After(q,\sigma)=\{q'\in Q\mid q=\sigma\Rightarrow q'\}$；

(4) $Sub(q)=\{q'\in Q\mid \exists\sigma\in A^*:\mid q=\sigma\Rightarrow q'\}$.

$Init(q)$ 表示状态 q 上的所有可观察动作，$Trace(q)$表示状态 q 上的所有可观察动作迹. 迹(traces)是 LTS 理论中的重要概念，它是系统可观察动作所组成的一个有限序列. A 上所有迹的集合用 A^* 表示. 如果 $\sigma_1,\sigma_2\in A^*$，那么 $\sigma_1\cdot\sigma_2$表示 σ_1和 σ_2的连接，用$|\sigma|$表示迹的长度. $After(q,\sigma)$表示状

态 q 经过动作迹 σ 后到达状态 q'，$Sub(q)$ 表示 T 存在一条从状态 q 到达状态 q' 的动作迹 σ.

定义 3.　$p \in Q$，$(p - \tau \rightarrow p'$ implies $p' \Downarrow)$ implies $p \Downarrow$.

$p \Downarrow$ 表示状态 p 收敛，$\Downarrow$ 有两个作用：(1) 用于检测一个标记变迁系统是否存在着不可观察的无限序列；(2) 在状态 p 执行一个可观察动作序列后，是否能到达另外一个状态 p'，也称为收敛于状态 p'. 相对应的，$p \Uparrow$ 表示状态 p 发散，$\Uparrow$ 是 $\Downarrow$ 的否定，通常用于检测一个标记变迁系统是否存在不可观察的无限序列[16].

定义 4.　$p \in Q$，那么

(1) $p \Downarrow \varepsilon$，if $p \Downarrow$；

(2) $p \Downarrow s\, s'$，if $p \Downarrow$ and $(p = s \Rightarrow p')$ implies $(p' \Downarrow s')$.

规则(1)表明了 p 在不可见动作集合 τ 上收敛能够得出 p 也在空序列 ε(也就是没有任何动作)上收敛. 规则(2)定义了动作迹的可叠加性，这对于分析标记变迁系统的中间状态是非常有用的.

下面给出 LTS 的符号定义，如表 1 所示.

表　1

符号	定义
A	系统所有可观察动作集
a, b, c	集合 A 的元素
A^*	A 之上的字符串集
$\sigma, s, s', s_1, \cdots$	集合 A^* 的元素
ε	空序列
τ	系统不可观察的内部动作
A_τ	$A \cup \{\tau\}$
$\mu, \mu_1, \mu_2 \cdots$	集合 A_τ 的元素
$p, q, p_1, p_2 \cdots$	集合 Q 中的元素
$p - \rightarrow q$	$p - \tau \rightarrow q$
$p - \mu_1 \mu_2 \cdots \mu_n \rightarrow q$	存在 $p_1, \cdots, p_n, 0 < i < n$，满足 $p = p_0 - \mu_1 \rightarrow p_1 - \mu_2 \rightarrow \cdots \rightarrow p_{n-1} - \mu_n \rightarrow p_n = q$
$p - \mu_1 \mu_2 \cdots \mu_n \rightarrow$	存在 q，满足 $p - \mu_1 \mu_2 \cdots \mu_n \rightarrow q$
$p - \mu_1 \mu_2 \cdots \mu_n + \rightarrow$	$p - \mu_1 \mu_2 \cdots \mu_n \rightarrow q$ 的否定
$p = \Rightarrow q$ 或 $p = \varepsilon \Rightarrow q$	$p - \tau^n \rightarrow q$，$n \geq 0$
$p = \mu \Rightarrow q$	存在 p_1 和 p_2，满足 $p = \Rightarrow p_1 - \mu \rightarrow p_2 = \Rightarrow q$
$p = a_1 a_2 \cdots a_n \Rightarrow q$	存在 $p_1, \cdots, p_n, 0 < i < n$，满足 $p = p_0 = a_1 \Rightarrow p_1 = a_2 \Rightarrow \cdots \Rightarrow p_{n-1} = a_n \Rightarrow p_n = q$
$p = s \Rightarrow$	存在 q 使得 $p = s \Rightarrow q$
$p = s \not\Rightarrow$	$p = s \Rightarrow q$ 的否定

3.3　信任链规范说明状态集

我们将信任链一致性测试中使用的规范形式说明集合记为 $SPECS$，将规范实现的集合记为 $IMPS$. $SPECS$ 通常是对信任链实现的一种抽象定义，一般采用非形式化或者自然语言进行描述，系统实现的过程又是多种多样的. 因此，所谓的“一致性”就是 $IMPS$ 与 $SPECS$ 之间的一种关系[6]，记为 imp：$imp \subseteq IMPS \times SPECS$，任何 $(i, s) \in imp$ 或 $i\ imp\ s$ 表示 i 是 s 的一种一致性实现. 规范说明和规范实现都可以看作是一个标记变迁系统，即认为规范实现 i 与规范说明 s 都有相应的 $LTS(i)$ 和 $LTS(s)$.

我们在 3.1 节对信任链的建立过程进行了刻画，RTM 是构建信任链的主要实体，$Booting(m)$ 是对建立信任链的抽象描述，但是并未对动作、状态以及两者之间的关系进行细致刻画. 虽然信任链规范给出了具体的度量流程，比如建立度量 log、对度量事件做 extend 操作等，但由于 RTM 的宿主是 BIOS，RTM 一方面要实现系统的完整性度量，另一方面，RTM 同时也要和 BIOS 其余部分，和 System 进行交互，这就带来了两个问题：(1) RTM 建立信任链的方式可以不一样，(2) RTM 和其余实体的交互，系统自带的动作对信任链的建立过程带来了影响.

信任链的主要功能之一就是确保 System 中所有组件完整性并建立完整性度量日志. 图 3 描述了在建立信任链的过程中，RTM 的度量动作集与 System 启动动作集之间的交互关系，度量动作序列反映在 TPM 内部的 PCR 寄存器中，而 System 启动动作是计算机系统的组成部分，因此也影响着 RTM 度量的动作序列，比如在图 3 中，“Chassis Intrusion”动作使得机箱处于打开状态，该动作不是 RTM 能够控制的，而是 System 或者用户决定的，一旦该动作触发，RTM 就必须响应该动作，并改变在原动作迹中加入该动作，那么信任链的状态也就被改变了.

因此，为了能够刻画出 System 和 RTM 之间的交互关系，我们首先需要建立 System 动作集合和信任链状态之间的有向图，然后从测试的角度出发对该图进行约简，也就是除去影响信任链测试的内部动作和不可达状态，找出适合信任链规范测试的测试例集合，并生成测试用例，最后对信任链进行一致性测试.

首先，我们需要通过 LTS 对 System 动作集合和信任链状态进行建模，获得信任链规范说明对应的 $LTS(s)$.

根据信任链规范说明，图 4 描述了信任链建立过程中在 System 动作影响下的状态变迁关系. System 所有动作都被执行的情况下的信任链状态的标

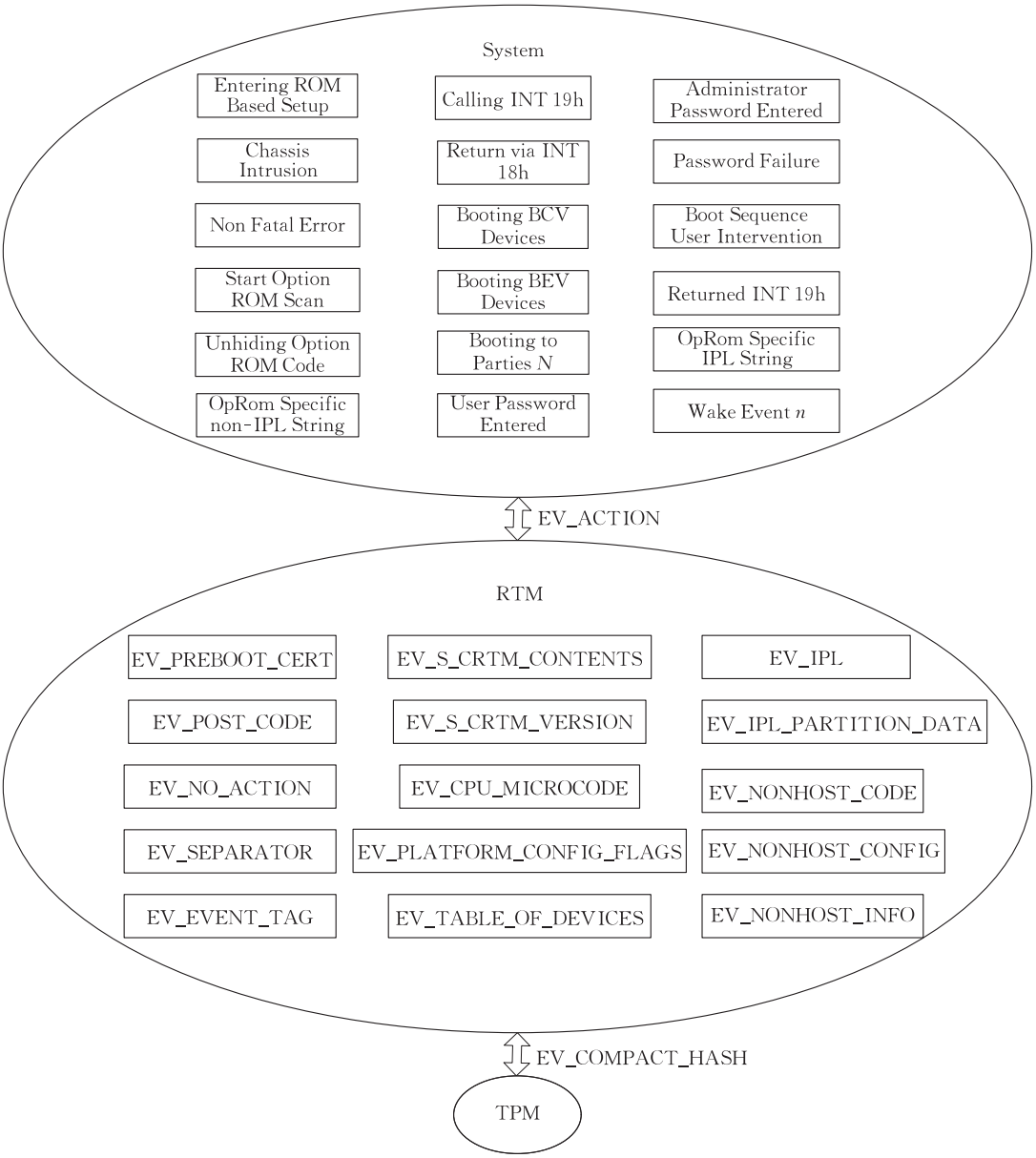

图 3 RTM 与 TPM，System 之间的交互关系

记变迁见附录. 我们在前面已经提到过，RTM 的度量动作可以通过完整性度量日志和 TPM 中的 PCR 寄存器反映出来，而 System 的动作集合却是不确定的，系统的运行状态和用户设定会改变该集合中的元素，为了能够准确地对信任链进行测试，就需要抽取出一套通用的 System 动作集，那么就能够保证信任链的可观察状态是确定的，也就是说是收敛的. 根据计算机系统启动的特点，状态 S_6 和 S_{15} 是系统必须经过的两个状态，状态 S_6 刻画了 BIOS 枚举并加载系统中的一个可启动设备，状态 S_{15} 则反映了系统开始运行操作系统加载器，并开始进入软件环境. 这里要注意的是，状态 S_0 和 S_{17} 并非是信任链的初始状态和终结状态，而是信任链与系统交互这段时间内的状态变迁，在此之前是可信度量根度量系统的过程，当到了状态 S_0 时，此时用户已经可以干预或者影响系统的行为，并且不同可信计算平台上设备的差异也导致信任链的建立序列将有所不同. 但对于测试人员来说，这些行为都是可观察的，同时也是可测试的. 图 4 中的 18 个状态描述了 System 运行态与信任链建立之间的交互过程.

$LTS(s)$ 的可观察动作集合 $A=\{a_i \mid 0\leq i\leq 17\}$，$Q\times Q$ 上的二元关系 R 总共有 43 种，若考虑从状态 S_0 到状态 S_{17} 并经过状态 S_6 和 S_{15} 的路径，则共有 864 条. 而在系统实际运行过程中出现的各种变化，加上规范说明中对信任链建立的约束条件，我们从易测试的角度出发[17]，图 4 所示的状态变迁关系实

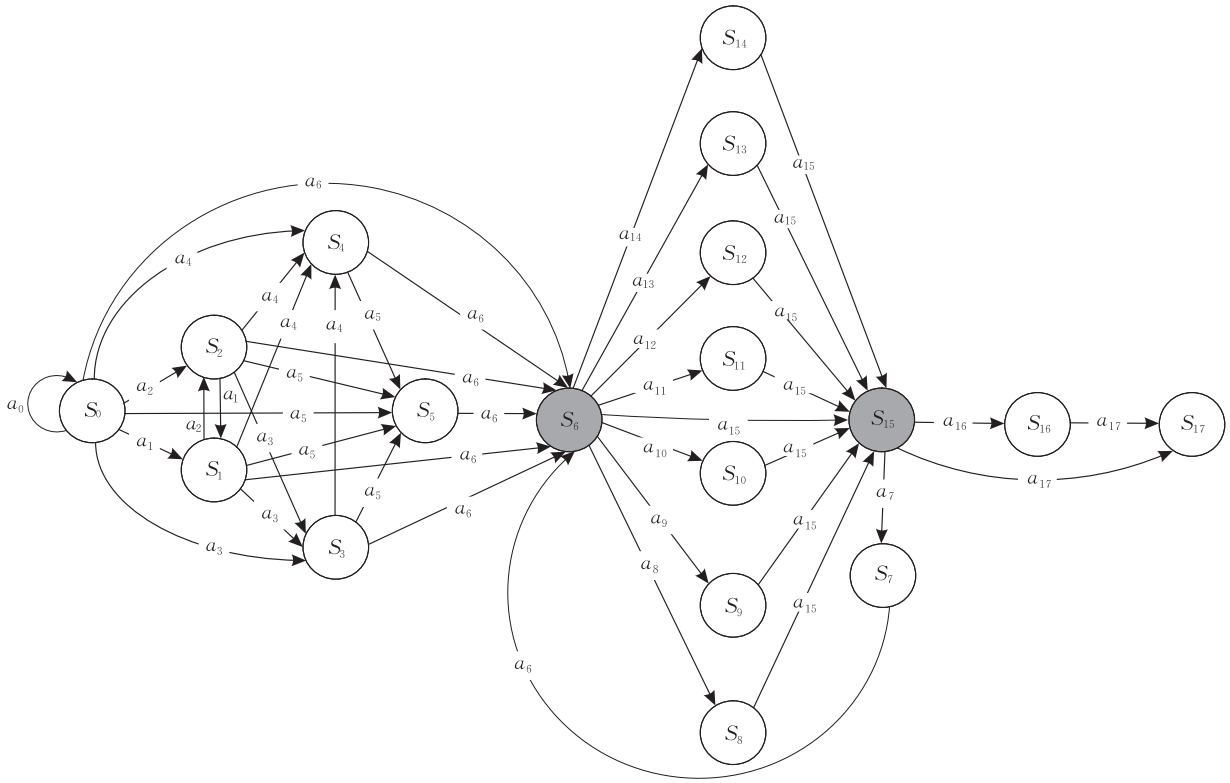

图 4　信任链 $LTS(s)$ 系统

际上是可以进行约简的.

3.4　信任链规范实现测试集

对于测试用例而言，为了保证它尽可能多地满足各个测试需求，而且数量少，就需要对测试目标中的所有测试需求进行整体优化，从而能生成最少的测试用例，实现对测试目标的充分测试[18]. 在可信计算 PC 规范说明中，对于 $LTS(s)$ 的可观察动作集合 A 是理想情况下所有动作的集合，在信任链规范实现过程中，有些动作是视情况而定的，譬如动作 a_1 和 a_2 属于规范说明中的可选动作，动作 $a_{8\sim14}$ 表示特殊启动设备（如网卡远程启动、SCSI 硬盘启动）或者用户操作（如输入 BIOS 开机密码、变更设备启动顺序），对于一般可信计算平台来说，这些动作属于多余动作，因此规范实现的 $LTS(i)$ 系统可以不考虑这些动作，而只去关注规范强制规定的动作集合. 因此约简规则可以表示为

(R1) $LTS(s)$ 中的状态 S_1 和 S_2 是可以被去除的；

(R2) $LTS(s)$ 中的状态 $S_8\sim S_{14}$ 是可以被去除的；

(R3) $LTS(s)$ 的起始状态是 S_0，结束状态是 S_{17}；

(R4) $LTS(s)$ 必须包含状态 S_6 和 S_{15}.

依次对 $LTS(s)$ 系统运用约简规则（R1）和（R2），并满足约简条件（R3）和（R4）. 经过约简后的 $LTS(i)$ 系统如图 5 所示，经过约简后，$LTS(i)$ 的可观察动作集合比 $LTS(s)$ 的少了一半，$Q\times Q$ 上的二元关系 R 减少至 17 种，从状态 S_0 到状态 S_{17} 并经过状态 S_6 和 S_{15} 的路径减少至 21 条.

为了能够比较 $LTS(s)$ 和 $LTS(i)$ 之间的关系，这里我们使用迹前序 $\leq_{tr}$ 进行评判；同时，由于我们

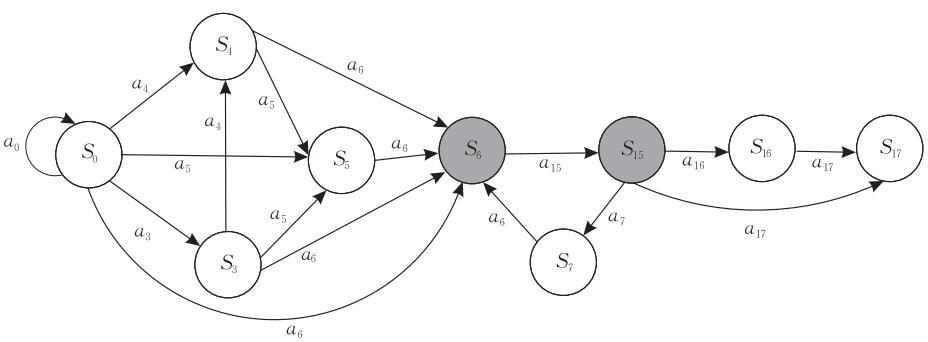

图 5　信任链 $LTS(i)$ 系统

采用了 $LTS(i)$ 进行信任链测试，为了能够说明 $LTS(i)$ 和被测系统平台 $LTS(i)_{tcp}$ 之间的关系，这里我们使用测试前序 $\leq_{te}$ 进行评判.

定义 5. 设 $i \in LST(i)$，$s \in LTS(s)$，则 $i \leq_{tr} s$ if $Trace(i) \subseteq Trace(s)$.

迹前序表示规范实现的 LTS 树是规范说明的 LTS 树的子树. 图 4 所刻画的是信任链规范说明中的 $LTS(s)$ 系统，而实际的信任链规范实现的 $LTS(i)$ 系统和前者应该是一种 $\leq_{tr}$ 二元关系的，也就是说 $Trace(i) \subseteq Trace(s)$，也就是图 5 所刻画的信任链 $LTS(i)$ 系统.

定义 6. $i \leq_{te} s$ iff $(\forall \sigma \in A^*, \forall a \subseteq A: After(i, \sigma - a)$ implies $After(s, \sigma - a))$.

测试前序和迹前序正好相反，$\leq_{te}$ 表示规范说明的 LTS 树是规范实现的 LTS 树的子树. 这是由于规范说明中满足 $After(s, \sigma - a)$ 的 $\sigma - a$ 使得规范实现中的 $After(i, \sigma - a)$ 不成立，也就是 $i \Uparrow (\sigma - a)$，这也反映了规范实现与规范说明之间的差异.

命题 1. $LTS(i)$ 是经过对 $LTS(s)$ 进行动作约简后得到的标记变迁系统，$LTS(i)$ 和 $LTS(s)$ 满足迹前序，即 $i \leq_{tr} s$.

证明. 根据定义 3 有 $i \leq_{tr} s$ if $Trace(i) \subseteq Trace(s)$，又根据定义 2 有 $Trace(q) = \{\sigma \in A^* \mid q = \sigma \Rightarrow\}$，因此只需证明 $\{\sigma_i\} \subseteq \{\sigma_s\}$.

根据命题条件：$LTS(i)$ 是经过对 $LTS(s)$ 进行动作约简后所得，因此有 $A_i \subseteq A_s$，那么有 $A_i^* \subseteq A_s^*$，而 $\sigma_i \in A_i^*$ 并且 $\sigma_s \in A_s^*$，因此有 $\{\sigma_i\} \subseteq \{\sigma_s\}$. 证毕.

命题 2. $LTS(i)$ 是经过对 $LTS(s)$ 进行动作约简后得到的标记变迁系统，$LTS(i_{tcp})$ 是任意一个可信计算平台信任链系统，$LTS(i)$ 和 $LTS(i_{tcp})$ 满足测试前序，即 $i \leq_{te} i_{tcp}$.

证明. 根据定义 6，证明 $i \leq_{te} i_{tcp}$ 等价为证明命题：$T = \{\forall \sigma \in A^*, \forall a \subseteq A \mid After(i, \sigma - a)$ implies $After(i_{tcp}, \sigma - a)\}$ 成立.

(1) 若 $After(i, \sigma - a)$ 使得 $i \Uparrow (\sigma - a)$，T 显然成立.

(2) 若 $After(i, \sigma - a)$ 使得 $i \Downarrow (\sigma - a)$，那么能够得到 $q = b \Rightarrow q'$，$q \Downarrow$ 和 $q' \Downarrow$，其中 $q \in Q_i$，$b = \sigma - a$. 因此只需证明 $\exists p \in Q_{i_{tcp}}$，使得 $p = b \Rightarrow p'$，$p \Downarrow$ 和 $p' \Downarrow$ 成立.

根据定义 3，$LTS(i_{tcp})$ 中没有无限不可观察动作序列，因此 $p \Downarrow$ 和 $p' \Downarrow$ 成立. 根据命题条件可知：$Trace(i) \subseteq Trace(i_{tcp})$，因此有 $A_i \subseteq A_{i_{tcp}}$，那么有 $A_i^* \subseteq A_{i_{tcp}}^*$，又 $b \subseteq A_i^*$，显然有 $b \subseteq A_{i_{tcp}}^*$.

若 $\neg \exists p' \in Q_{i_{tcp}}$，使得 $p = b \Rightarrow p'$，那么根据定义 3 和定义 4 得出 $p \Uparrow$，这与前面推出的 $p \Downarrow$ 矛盾，因此 $\exists p'$，使得 $p = b \Rightarrow p'$. 证毕.

命题 1 说明了规范实现 $LTS(i)$ 中的动作序列一定是规范说明 $LTS(s)$ 中的动作序列的子集.

命题 2 说明了系统 $LTS(i_{tcp})$ 中的状态集合包含或等于规范实现 $LTS(i)$ 中的状态集合，这反映了通过 $LTS(i)$ 生成的测试集合适合所有被测平台，$LTS(i_{tcp})$ 和 $LTS(i)$ 的状态补集反映出了对规范说明 $LTS(s)$ 中的动作约简，约简的动作要么是规范说明中的可选动作，要么是根据实际系统删除的特殊动作.

4 信任链测试系统

下面的工作是将信任链测试例和测试集进行形式化. 一个测试例是测试者与被测对象相互作用的行为的描述. 这种行为也可以表示成一个 LTS，但是测试要在有限时间内完成，因此测试行为以及描述测试例的 LTS 必须是确定且有限的. 同时我们要给测试例的每个结束状态一个判决(pass or fail). 下面是其定义：

(1) 测试例是一个五元组 $\langle Q, A, -\mu\rightarrow, v, s_0 \rangle$，其中 $\langle Q, A, -\mu\rightarrow, s_0 \rangle$ 是一个动作有限且确定的 LTS，v 是一个判决函数 $v: Q \rightarrow \{\text{fail}, \text{pass}\}$. 我们用 $LTS_t(A)$ 表示 A 上的所有测试例的集合. 而一个测试例可以看作是一个 LTS 的扩展.

(2) 测试套 T 是测试例的一个集合：$T \in P(LTS_t(A))$，这里 $P(LTS_t(A))$ 指 $LTS_t(A)$ 的幂集，即 $LTS_t(A)$ 的所有子集的集合.

测试例的运行可以看作是测试例与被测对象同步并行的相互作用直至不能继续作用下去为止或者说发生死锁[6].

这里的测试例是 3.4 节中所描述的规范实现标记变迁系统 $LTS(i)$，如果被测平台 S 能够通过测试套 T_i 的测试，那么根据命题 1，被测平台 S 中的所有全局动作都在 $LTS(s)$ 中，也就是说 S 中的所有可观察动作都满足 $LTS(i)$ 所定义的规则，也就是规范说明.

同时，$LTS(i)$ 是 $LTS(s)$ 的基本集，它刻画了一般意义下的可信计算平台系统与信任链的交互关系. 但 $LTS(i)$ 的路径个数仍然有 21 条，对于一个可信计算平台实例而言，很难同时满足这 21 种不同情况的测试. 为此，我们就需要在 $LTS(i)$ 中找到一个子图，我们称之为最大状态数测试系统 $LTS(i_m)$，$LTS(i_m)$ 满足如下条件：

(1) $LTS(i_m)$起始状态为 S_0，经过状态 S_6 和 S_{15}，至结束状态 S_{17}；

(2) $LTS(i_m)$中具有最大状态数且没有环路.

$LTS(i_m)$是 $LTS(i)$的一个实例，因为它包含了初态和终态，因此就能清楚地描述系统启动中信任链的建立过程，$LTS(i_m)$子图如图 6 所示.

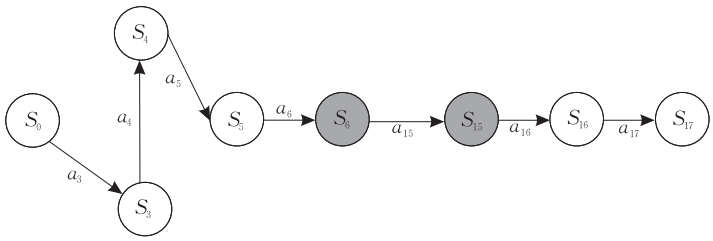

图 6　信任链 $LTS(i_m)$测试实例

构造出 $LTS(i_m)$之后就可以开始对可信计算平台信任链进行测试了，我们认为信任链模型的三个实体中，RTM 的作用是最重要的，因此除了对信任链系统进行 LTS 形式化分析之外，我们还对 RTM 的基本功能属性和完整性度量属性进行了测试，包括平台证书、支撑软件、对外接口和事件日志，但这些不是本文要讨论的重点. 不同可信 PC 平台的信任链测评结果如图 7 所示.

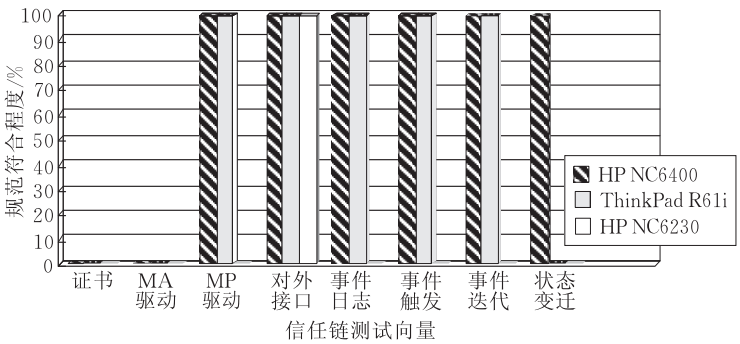

图 7　不同可信 PC 平台信任链测试结果

5　结　论

信任链是保证可信计算平台软硬件系统可信启动的一个必要环节，同时也是建立可信计算平台环境的基础，并且是平台远程证明、本地代码安全执行的先决条件. 因此对可信计算平台信任链进行规范一致性测试是非常有必要的.

然而，由于目前信任链机制本身尚缺乏理论模型，不同厂商对信任链的实现方式在细节上存在不同程度的差异，这两个因素也是制约对信任链进行有效测评的主要原因. 本文从系统易测试的角度出发，通过分析规范说明 $LTS(s)$和规范实现 $LTS(i)$之间的关系，归纳出信任链测试基本测试集，并从动态的角度建立了基于进程代数的信任链测试模型，并对该模型进行了形式化描述和验证，提出一种系统实现与规范实现之间的一致性关系，以此为基础实现了可对不同可信计算平台信任链进行测试的原型系统.

通过对信任链系统进行动作细化和行为分析，分离出系统中影响信任链稳定工作的动作；运用观察等价发现不同平台或者不同域的信任链之间的等价关系；通过一致性测试对可信计算平台信任链进行安全性分析，这些是我们下一步准备研究的内容.

参　考　文　献

[1] Shen Chang-Xiang, Zhang Huang-Guo, Feng Deng-Guo et al. Survey of information security. Science in China Series: E (Information Security), 2007, 37(2): 129-150(in Chinese)
(沈昌祥，张焕国，冯登国等. 信息安全综述. 中国科学 E 辑，2007，37(2)：129-150)

[2] Zhang Huan-Guo, Luo Jie, Jin Gang et al. Development of trusted computing research. Journal of Wuhan University (Natural Science Edition), 2006, 52(5): 513-518(in Chinese)
(张焕国，罗捷，金刚等. 可信计算研究进展. 武汉大学学报(理学版)，2006，52(5)：513-518)

[3] Sailer R, Zhang Xiao-Lan, Jaeger T et al. Design and implementation of a TCG-Based integrity measurement architecture//Proceedings of the 13th Usenix Security Symposium. San Diego, California, 2004: 223-238

[4] Wu Shi-Zhong. Decade research of testing, evaluation and

certification of information security. Information Security and Communications Privacy, 2007, 6(4): 5-8(in Chinese)
(吴世忠. 信息安全测评认证的十年求索. 信息安全与通信保密, 2007, 6(4): 5-8)

[5] ISO/IEC 9646-1. Information technology-open systems interconnection-conformance testing methodology and framework-part 1: General concepts. 1994

[6] Jiang Fan, Ning Hua-Zhong. Automatic test suite generation based on labeled transition system. Journal of Computer Research and Development, 2001, 38(12): 1435-1445(in Chinese)
(蒋凡, 宁华中. 基于标号变迁系统的测试集自动生成. 计算机研究与发展, 2001, 38(12): 1435-1445)

[7] Zhao Rui-Lian, Min Ying-Hua. A software testing method based on domain analysis of specification and program. Journal of Computer Research and Development, 2003, 40(6): 846-855(in Chinese)
(赵瑞莲, 闵应骅. 一种基于规范和程序域分析的软件测试方法. 计算机研究与发展, 2003, 40(6): 846-855)

[8] Keller R M. Formal verification of parallel programs. Communications of the ACM, 1976, 19(7): 371-384

[9] Lien Y E. Study of theoretical and practical aspects of transition systems [Ph. D. dissertation]. University of California, Berkley, 1972

[10] Plotkin G. A structural approach to operational semantics. Department of Computer Science Research Report DAIMI FN-19, Aaxhus University: Technical Report DAIMI FN-19, 1981

[11] Wang Yong-Xiang, Wu Jin-Zhao, Jiang Jian-Min. Process Algebra—Symmetrical and Action Refinement. Beijing: Science Press, 2007(in Chinese)
(王永祥, 吴尽昭, 蒋建民. 进程代数—对称与动作细化. 北京: 科学出版社, 2007)

[12] Li Meng-Jun, Li Zhou-Jun, Chen Huo-Wang. A survey of security protocol verification based on process algebra. Journal of Computer Research and Development, 2004, 41(7): 1093-1103(in Chinese)
(李梦君, 李舟军, 陈火旺. 基于进程代数安全协议验证的研究综述. 计算机研究与发展, 2004, 41(7): 1093-1103)

[13] Bi Jun, Wu Jian-Ping, Zhu Jiang. Research on extended transition systems based conformance testing for distributed systems. Acta Electronica Sinica, 2000, 28(2): 130-132(in Chinese)
(毕军, 吴建平, 朱江. 基于扩展变迁系统的分布式系统一致性测试研究. 电子学报, 2000, 28(2): 130-132)

[14] Deepak G, Jason F, Dilsun K et al. Towards a theory of secure systems. Technical Reports CMU-CyLab-08-003, 2008

[15] Bolognesi T, Brinksma E. Introduction to the ISO specification language LOTOS. Computer Networks and ISDN Systems, 1987, 14(1): 25-59

[16] Nicola R D. Extensional equivalences for transition systems. Acta Informatica, 1987, 24(2): 211-237

[17] Zhao Liang, Wang Jian-Min, Sun Jia-Guang. Study on the relationship between software testability and reliability. Chinese Journal of Computers, 2007, 30(6): 986-992(in Chinese)
(赵亮, 王建民, 孙家广. 软件易测性和软件可靠性关系研究. 计算机学报, 2007, 30(6): 986-992)

[18] Nie Chang-Hai, Xu Bao-Wen. A minimal test suite generation method. Chinese Journal of Computers, 2003, 26(12): 1690-1695(in Chinese)
(聂长海, 徐宝文. 一种最小测试用例集生成方法. 计算机学报, 2003, 26(12): 1690-1695)

附录.

附表 1 信任链 *LTS*(*s*)标记变迁关系

$P-\mu\rightarrow Q$	P	Q	μ
$S_0-a_0\rightarrow S_0$	S_0	S_0	进入 BIOS ROM Setup 界面
$S_{0,2}-a_1\rightarrow S_1$	$S_{0,2}$	S_1	机箱被打开
$S_{0,1}-a_2\rightarrow S_2$	$S_{0,1}$	S_2	系统启动中产生的非致命错误
$S_{0\sim2}-a_3\rightarrow S_3$	$S_{0\sim2}$	S_3	BIOS 扫描 Option ROM
$S_{0\sim3}-a_4\rightarrow S_4$	$S_{0\sim3}$	S_4	度量 Option ROM 的内部代码
$S_{0\sim4}-a_5\rightarrow S_5$	$S_{0\sim4}$	S_5	度量非引导类型的 Option ROM
$S_{0\sim5,7}-a_6\rightarrow S_6$	$S_{0\sim5,7}$	S_6	调用系统中断 INT 19h
$S_{15}-a_7\rightarrow S_7$	S_{15}	S_7	BIOS 通过 INT18h 重新获得控制权
$S_6-a_8\rightarrow S_8$	S_6	S_8	BIOS 启动一个 BCV(引导连接向量)设备
$S_6-a_9\rightarrow S_9$	S_6	S_9	BIOS 启动一个 BEV(引导项向量)设备
$S_6-a_{10}\rightarrow S_{10}$	S_6	S_{10}	BIOS 启动一个隐藏的 Parties 分区
$S_6-a_{11}\rightarrow S_{11}$	S_6	S_{11}	用户输入了正确的开机密码
$S_6-a_{12}\rightarrow S_{12}$	S_6	S_{12}	用户输入了正确的管理员开机密码
$S_6-a_{13}\rightarrow S_{13}$	S_6	S_{13}	用户输入了错误的开机密码
$S_6-a_{14}\rightarrow S_{14}$	S_6	S_{14}	用户改变了启动顺序
$S_{6,8\sim14}-a_{15}\rightarrow S_{15}$	$S_{6,8\sim14}$	S_{15}	进入中断调用 INT 19h 处理程序
$S_{15}-a_{16}\rightarrow S_{16}$	S_{15}	S_{16}	度量引导类型的 Option ROM
$S_{15,16}-a_{17}\rightarrow S_{17}$	$S_{15,16}$	S_{17}	某个事件唤醒主机加电

XU Ming-Di, born in 1980, Ph. D. candidate. His research interests include trusted computing and system security.

ZHANG Huan-Guo, born in 1945, professor. His research interests include information security and trusted computing.

YAN Fei, born in 1980, lecturer. His research interests focus on trusted computing.

Background

This research work is based on the authors' research direction which aimed at testing and evaluation of Trusted Computing Platform.

At present, the Trusted Computing has become a major development trend in the field of computer security. China's Trusted Computing technology and industry is in a phase of vigorous development, with the Trusted Computing product maturity, the Trusted Computing Platform applications will become more and more widely. Trusted Computing Platform can be used to greatly enhance the security of information systems, but must be under the Trusted Computing Platform evaluation, otherwise, neither the quality of trusted computing products, nor the security of information systems can be guaranteed.

According to "China National Information Security Certification and Evaluation Management", the information security products must be evaluated and certified. Testing and evaluation of Trusted Computing Platform lays emphasis on scientific, comprehensive, accurate testing and evaluation of the consistency and safety of Trusted Computing Platform product, and to ensure that the results is a comprehensive, scientific and internal consistency one. The main features of Testing and evaluation of Trusted Computing Platform includes standards or specifications conformance testing, safety testing and system reliability testing of Trusted Computing Platform. At present, there is no complete theory, technology and evaluation system for testing and evaluation of the Trusted Computing Platform at home and abroad.

This paper carries out testing a TCG-enabled PC and finds the gap between TCG specifications and product implementations. The authors used process algebra as a formal tool for description of the trust chain to deduct the behavior features of it, constructing an testing model of trust chain based on labeled transition system from the dynamic point of view, getting a reduction of basic testing set from testability point of view, proposing the consistency relations between implementation and specification of trust chain.

After analyzing characteristics of the trust chain, dividing testing of the trusted chain into CRTM testing, events testing and the state testing, putting forward a trusted Chain testing model constructed by the three kinds of testing mentioned above. The prototype aim at testing TCG-compliant products which are specific desktops or laptops, and delaminating testing model into three layers which are composed of TPM, chain of trust and TSS. According to experiments, there are some limitations in those products which had got rid of important functions such as event logs, MA driver. The vendor's convenient actions maybe destroy the original scheme designed by TCG.

This paper is supported by the National High Technology Research and Department Program (863 Program) of China (2007AA01Z411, 2008AA01Z404), National Natural Science Foundation of China (60673071, 90718005, 90718006) and the Open Fund of the Key Laboratory of Aerospace Information Security and Trust Computing, Ministry of Education.